

PATENT  
104108-0014

1C490 U.S. PRO  
09/606582  
06/29/00



**UNITED STATES PATENT APPLICATION**

*of*

**Michael A. Falco**

*for*

**RTP-FORMATED MEDIA CLIPS**

## RTP-FORMATED MEDIA CLIPS

### BACKGROUND OF THE INVENTION

#### *Field of the Invention*

The present invention is directed to persistent data storage and in particular to persistent storage of real-time data.

#### *Background Information*

A wide variety of media have been employed for storing real-time-type data persistently. In addition to the media traditionally employed to store such information in an analog manner, compact disks, digital versatile disks, and magnetic disks are all currently employed to store such data digitally. Because real-time data tends to be voluminous, a great degree of effort has been dedicated to storing it digitally in a format that makes data storage and use economical and convenient.

As a consequence, persistent real-time-data storage has achieved a high degree of sophistication and effectiveness.

### SUMMARY OF THE INVENTION

But I have recognized that a further advance in such storage will greatly contribute to the ease with which such storage and the resultant playback can be performed. Specifically, in storing real-time data, I format the data as payloads of RTP packets, i.e., as payloads of packets substantially of the type described in the Internet community's Request for Comments 1889 ("RFC 1889"). The payloads are stored with the RTP header's accompanying timestamp information so

as to specify the data's relative playback timing explicitly rather than only implicitly, from their positions in the data stream.

Since real-time data tend to be transmitted and received in RTP packets, such storage makes it quite convenient and computationally inexpensive to store 5 received real-time information and to transmit such information when it has been retrieved from storage. Additionally, the fact that the RTP format lends itself to separate storage of video and audio information yields significant versatility to such data's playback.

10

## BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

Fig. 1 is a block diagram that illustrates a number of environments in which the present invention's teachings can be employed;

15

Fig. 2 is a format diagram depicting one format environment in which an RTP packet may be transmitted;

Fig. 3 is a format diagram of an RTP header; and

Fig. 4 is a format diagram of a data segment employed by one embodiment of the present invention to store data persistently.

20

## DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

Figure 1 is a block diagram that is representative of a number of ways in which the present invention's teachings can be practiced. Figure 1 includes a block 12 representing some type of sampler, such as a video camera or microphone. This sampler creates a digital representation of a sound record and/or a 25 time-dependent scene, which a transmitter 14 may send over channel 16 to a receiver 18 for rendering by some kind of a player 20, such as speakers and/or a computer monitor. In one application, the channel 16 may be a packet network, in which case the receiver 18 would include an interface to that channel 16.

- The transmitter and receiver will typically be implemented in personal-computer or similar processors, which operate in accordance with stored programs contained in respective persistent stores 22 and 24, typically magnetic disks. Operating in accordance with this programming, such processors can
- 5 format data for storage in accordance with the present invention's teachings and act as drivers that operate the players 20 to play back data thus stored. In one application of the present invention's teachings the receiver 18 may record the received information in the persistent store 24 instead of, or contemporaneously with, driving the player 20 with that information.
- 10 So, although the drawing's legend for block 18 is "receiver," that block not only receives packets but also operates the persistent store 24 and drives the players 20. Similarly, the "transmitter" 14 performs several functions in addition to transmitting.

The transmitter 14 will typically break up the sampler 12's output sequence into discrete packets according to the RFC 1889 specification, so the receiver 18 will receive the real-time information in the form of RTP packets. Figure 2 is a format diagram that places such a packet in a typical, Ethernet environment, although RTP packets may be sent in other environments, too. If the receiver receives the packet over an Ethernet link, the RTP packet will be included in an Ethernet frame that starts with an Ethernet header and ends with an Ethernet trailer, as Fig. 2 illustrates. A typical Ethernet link may be shared by a number of nodes, and the Ethernet header indicates, among other things, which of the several nodes is to receive the Ethernet frame's contents.

If the Ethernet frame indicates that, say, Figure 1's receiver 18 is the proper recipient node, that receiver further consults the Ethernet header to determine which of its processes is to receive the Ethernet frame's payload. In the illustrative scenario, that process is the Internet Protocol ("IP") process, whose purpose is to determine whether—and, if so, how—to forward the received information along other links to which the recipient node may be coupled.

The IP process interprets the first part of the Ethernet frame's payload as an IP header, as Figure 2 indicates. The IP header includes an indication of the intended ultimate internetwork node for which the IP packet's payload is intended. For present purposes, we will assume that the intended internetwork 5 node is Figure 1's receiver 18 itself, i.e., that the receiver is not to forward it further but instead is to use the IP packet's payload itself.

The receiver 18 thereupon further inspects the IP header for an indication of the process to which the IP payload's contents should be delivered. In the illustrated scenario, that process is the User Datagram Protocol ("UDP") process. 10 The UDP process is the IP suite's transport-level protocol for delivering without providing reliability services. Its header, like the previous headers, includes de-multiplexing information to identify the process that can appropriately interpret the UDP datagram's payload.

In this case, that process is the RTP process. That process interprets the 15 UDP datagram's payload as beginning with an RTP header, whose format Figure 3 illustrates. The purposes of the various fields depicted in that drawing are well known to those skilled in the art, so we mention only the timestamp field here. That field is the one that specifies the relative timing of the samples contained in the RTP packet. The timestamp value does not represent an actual, 20 "wall-clock" time. Indeed, the value to be associated with the initial sample transmitted in an RTP session is chosen randomly. But the values after that represent the relative real times at which subsequent samples were taken.

The time duration represented by a single increment of the timestamp value depends on the particular type of data stream being transmitted, but the 25 receiver typically infers that value from the contents of the RTP header's PT field, which is a code for the type of data that the RTP payload represents. If, as Figure 2 indicates for the sake of example, the RTP payload is of the H.261 format often used to send video information, the timestamp would typically give the nominal sample time for all of the data within the packet, because the data typically all apply to the same video frame. In cases in which the RTP payload in-

stead contains audio information, not all of the data will typically have been taken at the same time. But they will represent an uninterrupted sequence of samples, and the timestamp value represents the first sample's time.

To play the information thus received, the receiver 18 drives the player 20  
5 with this information at times that correspond to the timestamps that accompany the information. That is, it establishes a relation between the advancing timestamp values and values on a local real-time clock, and it does not apply data to its player until its local wall-clock time reaches the time that corresponds to the timestamp value.

10 Now, although the transmitted information is ordinarily to be played in the fashion just described, the receiver 18 may instead, or additionally, record the transmitted information in persistent storage. In accordance with the present invention, the format with which real-time information is recorded is largely that set forth in the above-mentioned RTP specification. Specifically, the file containing  
15 the real-time information includes timestamp-containing headers to indicate the relative timing with which the various stored information is to be played.

A typical file for containing such information may, for instance, consist of a number of segments of which each has a format similar to that of Figure 4. Each such segment begins with a length field, which indicates how long the remainder  
20 of the segment is. In the case of a device, such as receiver 18, that has received RTP-format information delivered in a UDP datagram, this length field can be the value obtained from the UDP header. The next field includes at least an RTP timestamp, and it is designated the "RTP Header" field in Fig 4 because it typically will simply be a received packet's RTP header. Still, some implementations  
25 may omit certain RTP-header fields, such as those containing contributing-source identifiers, whose values may not be needed for the use to which the data will be put. Some implementations may additionally omit fields that were received with values different from those they have to contain during retransmission.

In any event, when the information is to be played or transmitted, the system for doing so plays or transmits it accordance with the timestamps. That is, it reads the timestamp for a given segment and then does not play or transmit it until the wall-clock time that the timestamp value dictates. If the system retransmits the stored media packets, it must replace the stored timestamp with a value consistent with the sender's wall clock as specified in RFC 1889. It must also replace the received synchronization-source identifier with one appropriate for the sender. In some cases the stored data will follow other data in the same session, and it will be preferable in such situations for the stored sequence number to be so modified as to indicate a discontinuity in the current media stream.

A difference greater than one between successive sequence numbers is such a discontinuity indicator.

The present invention's teachings will often be employed in multimedia situations, in which associated audio information is to be played with video information. As those skilled in the art will recognize, audio and associated video are usually transmitted in separate RTP sessions, and the present invention's implementations will typically store the information in that fashion. For instance, suppose that a multimedia presentation entitled "Scene 1" is to be stored. The video information may be stored in RTP format in a file called, say, "scene 1.261," where the ".261" extension may indicate that the format of the file's information is that specified by the International Telecommunications Union's H.261 standard. The audio may be stored in a file called, for instance, "scene 1.711" to indicate that it conforms to the G.711 audio-data standard.

Now, the timestamp information contained in the video file will not in general correspond to that in the audio file. As was mentioned above, timestamp values are assigned randomly at the beginning of a session, and the video and audio information will have been transmitted in separate sessions. Moreover, the time-interval length represented by one increment of the video timestamp values will not in general be the same as the length that an audio-timestamp-value in-

rement represents. It may therefore be considered desirable in some cases to add information that represents the relative timing between the two files.

But most implementers will not find this necessary. Ordinarily, the first samples from two related files can be counted on to have been taken close enough in time to be considered simultaneous, so timing can be based on this assumption. Additionally, the time-interval length represented by a single time-stamp-value increment may be inferred from, say, the file-name extension.

The present invention's approach to persistent storage lends itself to a wide variety of uses. As was just explained, a receiver can use it to keep a record of received information. For example, a receiver of a so-called voice-over-IP signal or of a videoconference signal may use the present invention to record the substance of a conference or other telephone call. This is clearly a simple way to store information that has been received in RTP format.

But the present invention is also advantageous for storing information that has not yet been transmitted. Figure 1's transmitter 14 may place the samples received from the sampling equipment 12 into the RTP format. Acting as a persistent-store driver, transmitter 14 stores the resultant packets in store 22 and then retrieves them for subsequent transmission or playback on local equipment (not shown). Transmitter 14 may, for instance, be a video- and/or audio-clip server, which sends stored information on demand to client network nodes. Storage in the RTP format greatly simplifies this process.

Additionally, the type of separate audio and video storage to which the RTP format lends itself, together with the fact that such information tends to be sent in different RTP sessions, affords a significant degree of flexibility. As was stated above, a given storage directory may contain a file called "scene 1.261" to represent audio information and another called "scene 1.711" to represent the related video information. The same directory may include a further file called, say, "scene 1.723," containing RTP information whose payload conforms to a different audio standard, which some clients may prefer. So the present invention makes it convenient to store different audio versions for the same video clip,

and vice versa. This applies not only to storage formats but also to, say, languages. For example, German, French, and Spanish versions of the audio for a film can be stored in different files and used with the same video file.

- This mechanism also provides a simple, efficient method to insert stored
- 5 media clips into a real-time media stream. A typical example of this function is to broadcast a message such as "Conference will end in five minutes" to all participants in a scheduled conference.

The present invention's teachings will also find considerable use for testing. Fig. 1 can be interpreted as illustrating a testing scenario. The transmitter 14 represents a test-signal source, which applies test signals to a unit under

10 test in the position of the channel 16 and receiver 18. For such purposes, the ability to employ a repeatable test signal is essential, and the RTP-format data in persistent store 22 would dictate the repeatable values to be transmitted.

Dashed lines 26 enclose elements of test equipment for performing such a test.

15 In some cases, circuitry 28 for comparing the unit under test's output with the input would provide most of the desired functionality. The output signal could be compared simply with the input signal or with a signal determined from other values contained in RTP format in a persistent store.

From the foregoing description, it is apparent that the present invention's

20 teachings can be employed in a wide range of embodiments and thus constitute a significant advance in the art.

What is claimed is: